MODIS Level 2 Corrected Reflectance Science Processing Algorithm

crefl_SPA

General

The NASA Goddard Space Flight Center's (GSFC) Direct Readout Laboratory (DRL), Code 606.3 developed this wrapper software for the National Polar-orbiting Operational Environmental Satellite System (NPOESS) Preparatory Project (NPP) In-Situ Ground System (NISGS) and the International Polar Orbiter Processing Package (IPOPP).

Users must agree to all terms and conditions in the Software Usage Agreement on the DRL Web Portal before downloading this software.

Software and documentation published on the DRL Web Portal may occasionally be updated or modified. The most current versions of DRL software are available at the DRL Web Portal:

http://www.directreadout.sci.gsfc.nasa.gov

Questions relating to the contents or status of this software and its documentation should be addressed to the DRL via the Contact Us mechanism at the DRL Web Portal:

http://directreadout.sci.gsfc.nasa.gov/index.cfm?section=contact%20us

Algorithm Wrapper Concept

The DRL has developed an algorithm wrapper to provide a common command and execution interface to encapsulate multi-discipline, multi-mission science processing algorithms. The wrapper also provides a structured, standardized technique for packaging new or updated algorithms with minimal effort.

A Science Processing Algorithm (SPA) is defined as a wrapper and its contained algorithm. SPAs will function in a standalone, cross-platform environment to serve the needs of the broad Direct Readout community. Detailed information about SPAs and other DRL technologies is available at:

http://directreadout.sci.gsfc.nasa.gov/index.cfm?section=technology

Software Description

This DRL software package contains the MODIS Corrected Reflectance Science Processing Algorithm (crefl_SPA). The crefl_SPA creates the MODIS Level 2 Corrected Reflectance product. The crefl_SPA performs a simple atmospheric correction with MODIS visible and near-infrared bands (bands 1 through 7). It corrects for molecular (Rayleigh) scattering and gaseous absorption (water vapor and ozone) using climatological values for gas contents. It requires no real-time input of ancillary data. The algorithm performs no aerosol correction. The Corrected Reflectance products created by crefl_SPA are very similar to the MODIS Land Surface Reflectance product (MOD09) in clear atmospheric conditions, since the algorithms used to derive both are based on the 6S Radiative Transfer Model. The products show differences in the presence of aerosols, however, because the MODIS Land Surface Reflectance product uses a more complex atmospheric correction algorithm that includes a correction for aerosols.

Software Version

Version 1 of the DRL algorithm wrapper was used to package the SPA described in this document.

Credits

The Corrected Reflectance algorithm (Version 1.4.2) was provided to the DRL by the MODIS Land Rapid Response Team.

Prerequisites

To run this package, you must have the Java Development Kit (JDK) or Java Runtime Engine (JRE) (Java 1.5 or higher) installed on your computer, and have the Java installation bin/ subdirectory in your PATH environment variable.

Program Inputs and Outputs

The MODIS L1B input products can be obtained from the DRL ftpsite. The datasets only cover the eastern US region.

DRL ftp site:

For Terra MODIS: ftp://is.sci.gsfc.nasa.gov/gsfcdata/terra/modis/level1/

For Aqua MODIS: ftp://is.sci.gsfc.nasa.gov/gsfcdata/aqua/modis/level1/

Datasets from other areas can be obtained either from the Distributed Active Archive Center (DAAC) or from other Direct Readout stations.

Output is the MODIS Level 2 Corrected Reflectance product.

Installation and Configuration

This section contains instructions for installing an SPA in a standalone configuration. SPAs may also be installed dynamically into an IPOPP framework; instructions for this type of installation are contained in the IPOPP User's Guide.

Download the crefl1.4.2_SPA.tar.gz and crefl1.4.2_SPA_testdata.tar.gz (optional) files into the same directory.

Decompress and un-archive the crefl1.4.2_SPA.tar.gz and crefl1.4.2_SPA_testdata.tar.gz (optional) files:

```
$ tar -xzf crefl1.4.2_SPA.tar.gz
$ tar -xzf crefl1.4.2_SPA_testdata.tar.gz
```

This will create the following subdirectories:

```
    SPA
        crefl
            algorithm
            ancillary
            station
            wrapper
            testscripts
            testdata
```

For convenience, this package contains binaries statically pre-compiled on an Intel-compatible 32-bit computer running under Fedora Core 6, using gcc 4.1.2. The binaries should work on most Linux OS/platforms. If you receive an error message while running the testscripts (refer to the next section, "Software Package Testing and Validation"), you may need to recompile the software for your platform/OS combination. Refer to the Appendix for instructions on recompiling the software.

Software Package Testing and Validation

The testscripts subdirectory contains a test script that can be used to verify that your current installation of the SPA is working properly, as described below. Note that the optional crefl1.4.2_SPA_testdata.tar.gz file is required to execute these testing procedures.

Step 1: cd into the testscripts directory.

Step 2: Run the 'run-crefl' script by typing: $./run-crefl

A successful execution usually takes some time (ranging from 5 to 10 minutes, depending on the speed of your computer), so if the execution seems to get stuck, do not become impatient. If everything is working properly, the scripts will terminate with a message such as:

Output modis.crefl is
/home/IPOPP/SPA/crefl/testdata/output/MODCREF.hdf

You can cd to the output directory to verify that the science product really exists. If it does exist, then the wrapped SPA works perfectly. Test output product(s) are available for comparison in the testdata/output directory. If there is a problem and the code terminates abnormally, the problem can be identified using the log files. Log files are automatically generated within the directory used for execution. They start with stdfile* and errfile*. Other problems may be caused by incompatibility between your system and the binaries provided with this software package. In that case you may need to recompile the software for your platform/OS combination. Refer to the Appendix for instructions on recompiling the software. Please report any errors that cannot be fixed to the DRL.

Program Operation

In order to run the package using your own input data, you can either use the 'run' scripts within the wrapper subdirectories, or modify the test scripts within the testscripts subdirectory.

To Use the Run Scripts

Identify the 'run' scripts: The wrapper/crefl directory contains the 'run' script. Execute the 'run' as described below to execute the crefl_SPA. Note that to execute 'run', you must have java on your path.

Specify input parameters using <label value> pairs: To execute the 'run' scripts, you must supply the required input and output parameters. Input and output parameters are usually file paths. Each parameter is specified on the command line by a <label value> pair. Labels are simply predefined names for parameters. Each label must be followed by its actual value. Each SPA has its own set of <label value> pairs that must be specified in order for it to execute. Some of these pairs are optional, meaning the process would still be able to execute even if that parameter were not supplied. There are three kinds of label/value pairs that the MODIS crefl_SPA uses, as follows:

a) Input file label/values. These are input file paths. Values are absolute or relative paths to the corresponding input file.

b) Parameter label/values. These are parameters that need to be passed onto the SPA (e.g., output resolution).

c) Output file labels. These are output files that are produced by the SPA. Values are the relative/absolute paths of the files you wish to generate.

The following table contains a list of labels, and their descriptions, that are needed by the SPA.

Input Labels	Description
modis.mxd021km	MODIS 1km L1B Calibrated Geolocated Radiances HDF file (MOD021KM, MYD021KM)
modis.mxd02hkm	MODIS 500m L1B Calibrated Geolocated Radiances HDF file (MOD02HKM, MYD02HKM)
modis.mxd02qkm	MODIS 250m L1B Calibrated Geolocated Radiances HDF file (MOD02QKM, MYD02QKM)
Parameter Labels	**Description**
resolution (optional)	1km or 500m.
Output Label	**Description**
modis.crefl	Corrected Reflectance product

NOTE: The parameter 'resolution' need not be used. If it is not used, the default HDF output would have corrected reflectance for MODIS bands 1 and 2 in 250m resolution, and corrected reflectance for all other solar reflective bands in 500m resolution. If however resolution is specified as 1km or 500m, all output bands would be in the specified resolution.

Execute the 'runs': The following is an example of a command line to run the crefl_SPA from the crefl/testscripts subdirectory. You can run it from any directory of your choice, by using the correct paths to the 'run' script and your datasets.

```
$ ../wrapper/crefl/run \
    modis.mxd02qkm   ../testdata/input/MOD02QKM.07053161056.hdf  \
    modis.mxd02hkm ../testdata/input/MOD02HKM.07053161056.hdf  \
    modis.mxd021km ../testdata/input/MOD021KM.07053161056.hdf \
    modis.crefl ../testdata/output/MODCREF.hdf \
    resolution 1km
```

```
Output modis.crefl is
/home/IPOPP/SPA/crefl1/testdata/output/MODCREF.hdf
```

A successful execution of 'run' usually takes some time (around 5 minutes, depending on the speed of your computer), so if the execution seems to get stuck, do not become impatient. If execution fails, you will see an error message indicating the cause of failure (e.g., a file cannot be found, or a label cannot be recognized). Correct it and run again. If the problem has some other cause, it can be identified using the log files. Log files are automatically generated within the directory used for execution. They start with stdfile* and errfile*. Other problems may be caused by incompatibility between your platform/OS and the binaries provided with this software package. In that case you may need to recompile the software for your platform/OS combination. Refer to the Appendix for instructions on recompiling the software. The executions will create some temporary files (or symbolic links) in your execution directory. Delete them after the run.

To Use the Script in the testscripts Directory

One simple way to run the algorithms from any directory of your choice using your own data is to copy the 'run-crefl' script from the testscripts directory to the selected directory. Change the values of the variables WRAPPERHOME, L1HOME, and OUTPUTHOME to reflect the file paths of the wrapper directories and the input/output file paths, and then modify the input/output file name variables. Run the script to process your data.

Appendix

Instructions for Recompiling the crefl_SPA Software

If you are going to build the binaries in this package, you will need to install the Hierarchical Data Format (HDF) library, HDF4.2rX on your system. The HDF4.2rX library can be obtained in precompiled binary form at http://hdf.ncsa.uiuc.edu. Download the library for your platform and install it. Make sure to get the JPEG, ZLIB and SZIP libraries for your platform and place all of the library and include files under the lib/ and include/ directories within the HDF install directory. Then cd into the SPA/crefl/algorithm directory. Modify "Makefile" to use the correct C compiler and any additional compilation options that may be necessary. Edit the HDFHOME variable to point to the correct HDF install directory. First run "make clean" to remove any existing binaries. Next run "make" to build the binaries.